Doc Code: AP.PRE.REQ

PRE-APPEAL BRIEF REQUEST FOR REVIEW

	Docket Number (Optional)
	2095.001100/P3126US1

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to "Mail Stop AF, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450" [37 CFR 1.8(a)]	Application Number	Filed
on August 26, 2009	10/660,353	September 11, 2003
Signature	First Named Inventor	
	P. Anders I. Bertelrud	
Typed or printed name Jaison C. John	Art Unit	Examiner
	2192	Eric B. Kiss

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached sheet(s).
          Note: No more than five (5) pages may be provided.

I am the

☐ applicant/inventor.

☐ assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

☑ attorney or agent of record. 50,737
Registration number

☐ attorney or agent acting under 37 CFR 1.34.

Registration number if acting under 37 CFR 1.34 _____

Signature

Jaison C. John
Typed or printed name

(713) 934-4069
Telephone number

August 26, 2009
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

☑ *Total of 1 forms are submitted.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

P ANDERS I. BERTELRUD
THEODORE C. GOLDSTEIN

Serial No.: 10/660,353

Filed: SEPTEMBER 11, 2003

Title: PREDICTIVELY PROCESSING
TASKS FOR BUILDING SOFTWARE

Group Art Unit:	2192
Examiner:	ERIC B. KISS
Conf. No.:	5128
Atty. Dkt.:	2095.001100/P3126

REMARKS CONCERNING PRE-APPEAL BRIEF
REQUEST FOR REVIEW

**MAIL STOP APPEAL BRIEF-
PATENT**
Commissioner for Patents
P. O. Box 1450
Alexandria, VA 22313-1450

Sir:

Appellants submit the following remarks concerning the Pre-Appeal Brief Request for Review filed concurrently herewith. The following remarks show that there are clear errors in the Examiner's rejections.

The Examiner rejected claims 1-4, 6-13, 15-21, 23-29 and 31-37 under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 5,193,191 *(McKeeman)*. Additionally, the Examiner rejected claim 30 under 35 U.S.C. 103(a) as being unpatentable over *McKeeman* in view of U.S. Patent Publication No. 2005/0108682 *(Piehler)*. The Examiner imposed these rejections in the Final Office Action mailed May 26, 2009. The Examiner issued an Advisory Action on August 19, 2009 maintaining the rejection of the Final Office Action. As illustrated below, the Examiner's statements in the Final Office Action represent clear errors.

The Examiner's rejection of claim 1 is improper because *McKeeman*, as cited by the Examiner, fails to teach at least one of the claimed features. For example, claim 1 calls for initiating compilation of a file in a processor-based system <u>in advance of</u> a request from a user to compile the file. The Examiner argues that *McKeeman* teaches this feature because *McKeeman* teaches that recompilation "uses previously compiled code." *See* Final Office Action, p.3 (citing *McKeeman*, col. 11, ll. 44-61). The passage cited by the Examiner, however, is completely silent with respect to initiating compilation of a file in a processor-based system <u>in advance of</u> a request

from a user. Regardless of whether or not *McKeeman* teaches using previously compiled code for recompilation, *McKeeman* does <u>not</u> teach initiating compiling <u>in advance of</u> a request from a user, as called for in claim 1. In fact, *McKeeman* teaches that the recompilation is done *after* (not <u>in advance of</u>) a user initiates a compile. *See, e.g., McKeeman*, col. 5, ll. 8-18 (stating "[w]hen the developer has reached a point where he wishes to test the code he has written, *the compiler 11 is invoked.* The input to the compiler 11 is the source code text produced by the editor 10." (emphasis added)).

In the Advisory Action, the Examiner argues that the "subsequent compilation" in *McKeeman* corresponds to the "initiating [compiling] in response to determining that the file has been modified." *See* Advisory Action, p.2. To the extent it is the Examiner's position that the subsequent compilation in *McKeeman* teaches that <u>initiating compilation comprises compiling the file in response to determining that the file has been modified</u>, Appellants point out that the compiling action of claim 1 referred to here **is the same** compiling action in the claimed feature of "initiating compilation of a file in a processor-based system in advance of a request from a user to compile." The Examiner, however, seems to take the position that the **subsequent** compilation and the **previous** compilation in *McKeeman* both correspond to the <u>initiating compilation</u> feature in claim 1. As explained below, this position is untenable.

In the Advisory Action, the Examiner stated that "[b]ecause the previous compilation results for portions of the file that have not been modified are reused in subsequent compilation, the previous compilation may be reasonably interpreted as initiating compilation...." Advisory Action, p.2. In other words, the Examiner states that the previous compilation in *McKeeman* corresponds to initiating compilation of a file in a processor-based system <u>in advance of</u> a request from a user. However, the Examiner also takes the position that the subsequent compilation in *McKeeman* corresponds to <u>initiating compilation comprises compiling the file in response to determining that the file has been modified</u>. *See id.* As Appellants have pointed out, the <u>initiating compilation</u> feature recited twice in claim 1 is a <u>single</u> feature with multiple aspects. The Examiner, however, improperly attempts to apply **two separate** compilations (previous and subsequent) from *McKeeman* to the single feature of <u>initiating compilation</u>. A subsequent compilation and a previous compilation cannot both be the single compilation (made in advance of a user request **and** in response to determining the file has been modified). The Examiner's application of *McKeeman* is inconsistently applied to claim 1. For at least these reasons, *McKeeman* does not, and cannot, anticipate all the features of claim 1.

The Examiner also states that the "subsequent compilation" of code in *McKeeman* corresponds to the "initiating compiling in response to detecting a user request." *See* Advisory Action, p.2. Claim 1, however, does not recite such a feature. However, claim 1 indeed recites compiling the file in response to determining that the file has been modified and indicating a status for the compilation of the file in response to detecting the user request. Insomuch as the Examiner rejects features not present in the claims, the Examiner's arguments are moot.

Claim 1 calls for indicating a status of the compilation of the file in response to detecting the user request. The Examiner argues that *McKeeman* teaches this claimed feature. *See* Final Office Action, p.3 (citing *McKeeman*, col. 5, ll. 21-13). The Examiner, as in the previous Office Action, improperly characterizes col. 5, lines 21-23 of *McKeeman*. The Examiner states the cited passage of *McKeeman* teaches initiating compilation in response to determining that the file has been modified. This is incorrect. Rather, the cited passage teaches that, *after* the user initiates compilation (col. 5, lines 15-17), if a source text (module 12) has not been changed, then it is not recompiled.

> "When the developer has reached a point where he wishes to test the code he has written, the compiler 11 is invoked. The input to the compiler 11 is the source code text produced by the editor 10. There are typically a number of source code test buffers 12, one for each module of the application under development; according to one feature of the invention, those modules 12 which have not been changed or are not dependent upon changed code are not recompiled."
> *McKeeman*, col. 5, ll. 15-23.

When properly read in context, the cited passage in *McKeeman* does not teach that the status of the compilation is indicated in response to detecting a user input, as argued by the Examiner. The Examiner's arguments cannot be correct because the compilation in *McKeeman* has not yet taken place when the user request is received. In other words, *McKeeman* fails to teach or suggest indicating the status of the compilation in response to detecting a user input. As such, *McKeeman* does not, and cannot, teach the claimed feature of the status of the compilation is indicated in response to detecting a user input, as called for in claim 1.

Claim 1 also calls for compiling the file in response to determining that the file has been modified. Again, the Examiner cites *McKeeman* , col. 5, ll. 21-23, as teaching this claimed feature. *See* Final Office Action, p.3. However, as Applicants have stated in the discussion above, *McKeeman* compiles when the user (developer) decides to compile, not in response to a file modification, as called for in claim 1.

For at least the aforementioned reasons, claim 1 and its dependent claims are allowable.

For at least similar reasons, the remaining independent claims, and their respective dependent claims are also allowable.

Other claims are also allowable for additional features recited therein. For example, claim 24 calls for, *inter alia*, initiating processing of at least a portion of the modified source files (*i.e.*, one or more source files that have been modified) **before** receiving a request to process the modified files. The Examiner asserts that this feature is taught in *McKeeman* at column 11, lines 44-61. *See* Final Office Action, p.8. *McKeeman* fails to teach this feature. The cited passage describes reusing previously gathered information (such as compiled code) at recompilation if the source text has *not* changed. *See McKeeman*, col. 11, ll. 44-61. Thus, this passage describes that, when recompilation is initiated, the compiler will compile only the changed files and will not recompile the unchanged source text, thereby saving unnecessary computation. This passage, however, does <u>not</u> describe initiating processing of one or more of the <u>modified</u> source files **before** a request to process the one or more source files (i.e., in *McKeeman*'s case, a request to recompile the changed files) is received. The "recompilation" in *McKeeman* involves the reuse of previously compiled code derived from unchanged source text, and compiling only those source files that have been changed. Notably, the changed files in *McKeeman* are processed <u>after</u> the user initiates the request to recompile (col. 5, ll. 15-17). In contrast, claim 24 calls for initiating the processing of modified source files **before** receiving the request to process the modified source files. For at least this reason, claim 24 and its dependent claims are allowable.

The Examiner improperly characterizes column 11, lines 44-61 of *McKeeman*. At page 8 of the Final Office Action, the Examiner argues that the cited passage of *McKeeman* teaches initiating processing of at least a portion of modified source files before receiving a request to process the modified files, and then receiving the request to process at least one of the modified files. This is incorrect. Rather, the cited passage teaches that processing of modified source files is initiated at recompilation time, i.e., <u>after</u> a request to process the modified files.

For at least these reasons, claim 24 and its dependent claims are allowable.

Turning to at least some of the dependent claims, claims 2-3, 12-13, 18-21, and 32-35 all recite the production of an object code file after compilation is initiated. In contrast, *McKeeman* is directed to generating debugged source code, <u>not</u> object code, and teaches later use of a different compiler to generate object code (col. 5, ll. 48-57).

Other dependent claims are also allowable for claimed features recited therein. For

Remarks on Pre-Appeal Brief Request for Review
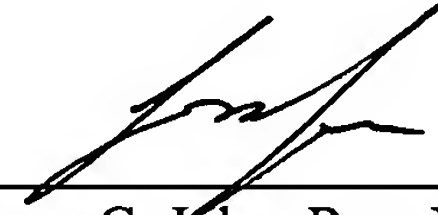Serial No. 10/660,353

example, claims 29 and 33 recite the use of at least one marker or at least two markers, respectively, in identifying a section of the source file that should be compiled. *McKeeman fails* to teach markers, let alone the use of markers in identifying a section of the source file that should be compiled.

In view of the foregoing, it is respectfully submitted that all pending claims are in condition for immediate allowance. The Examiner is invited to contact the undersigned attorney at (713) 934-4069 with any questions, comments or suggestions relating to the referenced patent application.

Respectfully submitted,

WILLIAMS, MORGAN & AMERSON, P.C.,
CUSTOMER NO. 23720

Date: <u>August 26, 2009</u>   By: _____
                                        Jaison C. John, Reg. No. 50,737
                                        10333 Richmond, Suite 1100
                                        Houston, TX 77042
                                        Tel: (713) 934-4069
                                        Fax: (713) 934-7011
                                        ATTORNEY FOR APPLICANT(S)